ANNOTATIONS TIME SHIFT: A KEY PARAMETER IN EVALUATING MUSICAL NOTE ONSET DETECTION ALGORITHMS

Mina Mounir,[1*] *Peter Karsmakers,*[2] *Toon van Waterschoot,*[1]

[1] KU Leuven, Dpt. of Electrical Engineering, ESAT-STADIUS/ETC, 3001 Leuven, Belgium
mina.mounir@esat.kuleuven.be, toon.vanwaterschoot@esat.kuleuven.be
[2] KU Leuven, Dpt. of Computer Science, TC CS-ADVISE, B-2440 GEEL, Belgium.
peter.karsmakers@kuleuven.be

ABSTRACT

Musical note onset detection is a building component for several MIR related tasks. The ambiguity in the definition of a note onset and the lack of a standard way to annotate onsets, introduce differences in datasets labeling, which in turn makes evaluations of note onset detection algorithms difficult to compare. This paper gives an overview of the parameters influencing the commonly used onset detection evaluation measure, i.e. the F1-score, pointing out a consistently missing parameter which is the overall time shift in annotations. This paper shows how crucial this parameter is in making reported F1-scores comparable among different algorithms and datasets, achieving a more reliable evaluation. As several MIR applications are concerned with the relative location of onsets to each other and not their absolute location, this paper suggests to include the overall time shift as a parameter when evaluating the algorithm performance. Experiments show a strong variability in the reported F1-score and up to 50% increase in the best-case F1-score when varying the overall time shift. Optimizing the time shift turns out to be crucial when training or testing algorithms with datasets that are annotated differently (e.g. manually, automatically, and with different annotators) and especially when using deep learning algorithms.

Index Terms— Musical note onsets, evaluation, acoustic event detection, machine learning

1. INTRODUCTION

Detecting the onsets of musical notes is like a hide-and-seek game in which we are trying to chase the starting of musical notes in a piece of music. Usually in those kind of games we have a good idea of what we hide and this moves the whole problem difficulty to the seeking part. For note onset detection this is not the case as literature provides a variety of definitions for a note start. For instance it could be either when a note is triggered or when it is perceived [1]. Considering a musical note as a sequence of a transient followed by a steady-state component [2], an onset is the point chosen to mark the transient [3] or more precisely it should be as close as possible to the transient's start [4]. But again a transient length depends on instrument and playing style and there is no objective way to measure how close is the onset to the transient's start.

Even if note onset detection is an established research problem, it is always capturing researchers' interest. On one hand, there is still quite some room for performance improvement. On the other hand, it plays a core role in a variety of music signal processing (adaptive audio effects [3], music synthesis [5]) and MIR applications (automatic music transcription [6], recommender systems [7] and music fingerprinting/search systems [8], [9], [10]).

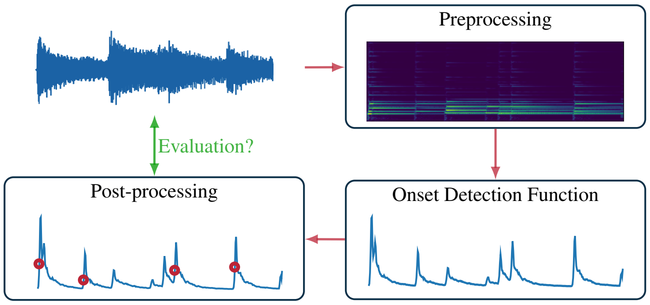

Figure 1: General scheme for note onset detection.

Literature is rich with many solutions proposed for detecting note onsets. Each solution starts by deciding on an onset definition which may depend on several factors: application, target instruments, available datasets, labeling method, availability of annotators, etc. The selected definition has a fundamental impact on how the ground-truth annotations are generated which may in turn drastically affect how the detections are evaluated. In many MIR applications like tempo estimation, songs search engines and music synthesis, algorithms make more use of onsets' relative positions rather than their absolute positions. On the other hand, for applications seeking exact onset times, the latter are generally defined with certain tolerance and adapted to the target application.

Before analyzing the performance evaluation, we first summarize the seeking part of the game. It usually follows a certain scheme of three steps [3] illustrated in Fig.1. Most of the research is focused on the middle step, trying to come up with a better *Onset Detection Function* (ODF) which is defined as a highly sub-sampled version of the input music signal presenting distinguishable amplitude peaks corresponding to onset locations. Existing ODFs are grouped in two main classes: probabilistic and non-probabilistic. Referring to MIREX results for the last years [11], the best performing state-of-the-art non-probabilistic algorithm was fluctuating between *ComplexFlux* [12] and *SuperFlux* [13] which are based on *LogSpecFlux*(LSF) [14][15], a method detecting onsets by spec-

[*] This research work was carried out at the ESAT Laboratory of KU Leuven. The research leading to these results has received funding from the KU Leuven Internal Funds C2-16-00449, IMP/14/037, and VES/19/004, and the European Research Council under the European Union's Horizon 2020 research and innovation program / ERC Consolidator Grant: SONORA (no. 773268). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

tral dissimilarity. Another non-probabilistic algorithm NINOS^2 [2], exploiting spectral sparsity, showed better results than LSF when applied to guitar chord progressions. The same performance is claimed to apply for polyphonic instruments where progressions share more harmonics which challenges LSF. For the probabilistic ODFs, deep learning algorithms were so far the best performing solutions in MIREX and the convolutional neural network (CNN) by [16] is considered to be the state-of-the-art solution [11].

When evaluating the ODFs using different datasets, one could expect that having a slight shift in the annotations between the training/tuning and testing datasets may result in a misleading performance assessment. This mismatch in labeling is quite common between different datasets due to many reasons, for instance:

- the lack of unified onset definition,
- datasets are labeled by different people and research groups,
- datasets may be of different nature (manually annotated by different annotators, synthesized from symbolic data, natural recording using a Yamaha Disklavier, mixing isolated notes).

In [4] authors suggest a methodology to manually label note onsets in a more consistent way. They also provided a dataset that was used by several researches, but unfortunately is not large enough for training a deep network. On the other hand, in [1], the authors focused on the evaluation parameters and measures. Referring to *Detection Performance* as the most important measure and analyzing the resolution of onset perception, the authors in [1] consider a correct detection to be within a time window of 50 ms around the reference onset time - 25ms on each of the annotation sides - which will be shown not enough when dealing with diverse datasets. In [17], it was shown how a minor labeling mismatch dramatically decreases the evaluation reliability for the melody extraction problem. In that paper, they were mainly considering the mismatch due to framing which is generally smaller than the inter-dataset annotation mismatch.

In this paper we introduce the *overall shift in annotations* parameter providing a means to overcome the datasets' labeling mismatch. We show how this parameter makes the evaluation metric less dependent on datasets and more focused on the performance of detection algorithms. Section 2 will give a concise overview of onset detection methods and will explain the parameters influencing the performance evaluation. The experiment environment and datasets will be summarized in Section 3. Finally Sections 4 and 5, will discuss the results and give some suggestions for future work.

2. ONSET DETECTION AND EVALUATION

2.1. Onset detection function

Considering the non-probabilistic ODFs, the logarithmic spectral flux (LSF) [14] is given by:

$$\mathit{LSF}(i) \;=\; \sum_{k=1}^{k=\frac{K}{2}} H(|Y_{ik}| - |Y_{i-1,k}|)\ , \tag{1}$$

where $H(x) = \frac{x+|x|}{2}$ is the half-wave rectifier function and $|Y_{ik}|$ is the logarithmic magnitude spectrogram for a music signal frame with frame index i, frequency bin k and K is the frame length. NINOS^2 ODF [2] is given by

$$\mathit{NINOS}^2(i) \;=\; \frac{\|Y_i\|_2^2}{\sqrt[4]{K}\|Y_i\|_4}\ , \tag{2}$$

calculating the inverse-spectral sparsity per frame which is high for transients and therefore for onsets. On the other hand the probabilistic CNN is trained on input points labeled with 0 for non-onsets and 1 for onsets. Each point is a 3D tensor formed by concatenating different spectrograms representing a short snippet of the signal. The network's output is treated as an ODF where thresholding is needed afterwards to decide on the output's class. As a consequence of being deep learners, added to the fact of being trained on binary labels, the CNN-ODF is quite different in its nature when compared the non-probabilistic ODFs. This is shown in Fig.2 with the dashed lines marking the onsets ground-truth. The CNN-ODF is more precise, i.e. it experiences a magnitude increase in a short window around the onset candidates which may increase evaluation sensitivity to labeling mismatch.

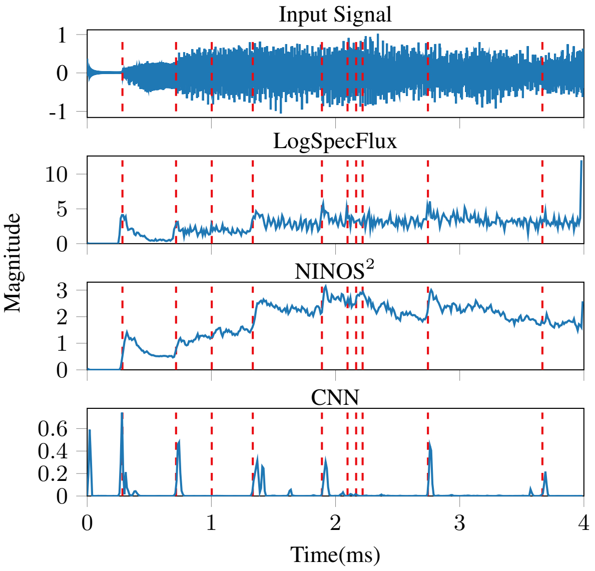

Figure 2: Comparison of LogSpecFlux, NINOS^2 and CNN ODFs for a *Cello* excerpt.

2.2. Peak-Picking

In general, the peak-picking starts by smoothing the signal in order to avoid glitches. Then comes either a *fixed* or an *adaptive* thresholding policy depending on the ODF's shape. For non-probabilistic techniques, researchers went to the adaptive choice, calculating a moving average of the ODF and picking the local maxima that are higher than this average with a certain threshold. This is described mathematically by detecting an onset at frame i only if the three following conditions hold:

1. $O(i) = \max_l\ O(i+l), \quad \text{with} \quad l = -\alpha, \ldots, +\beta,$
2. $O(i) \geq \frac{1}{a+b+1}\sum_{l=-a}^{+b} O(i+l) + \tau,$
3. $i - p > \theta,$

where (α, β) and (a, b) define the maximum and averaging windows respectively and with the last rule restricting the distance between any two onsets i and p to a minimum distance θ. The threshold τ is used when comparing the ODF to the moving average. The parameters α, β, a, b and τ are tuned together or separately for achieving the best performance while θ is heuristically fixed to a distance allowing on average two notes to be distinguishable and hence manually annotated. On the other hand, the CNN produces ODFs where magnitude peaks are centered around candidates' positions and are almost zero elsewhere. Picking all the maxima using only fixed thresholding will do the job reducing all the tunable parameters to the threshold τ. Both peak-picking ways, will produce a list of frame/input-point numbers which represent the detected onsets. Those frame numbers are usually translated then to time instants before feeding them to the evaluation process.

2.3. Evaluation method

For the evaluation we stick to the guidelines mentioned for the MIREX competition [11]. The F1-score is defined as the harmonic average of the *precision* (P) and the *recall* (R) given by:

$$F1 = \frac{2 \times P \times R}{P + R}, \tag{3}$$

with the precision being the ratio of correctly detected onsets to the total number of onsets under test, and the recall comparing the amount of correctly detected onsets to the total number of points detected as onsets. When reporting the F1-score, we should differentiate between two cases: best-case and on-average F1-score. Having a limited amount of annotated songs, many researchers are satisfied by sharing the highest achievable F1-score as a proof-of-concept. For this, an F1-score per threshold F1(τ) is calculated and averaged over all the test examples, and the highest F1-score corresponding to the optimal threshold τ^* is the reported best-case F1-score. Alternatively, an F1-score is reported for a hold out dataset using the optimal threshold τ^* calculated on the validation set which is similar to MIREX evaluation as the dataset is not public. For a more elaborate evaluation, an on-average F1-score is reported using a k-fold cross-validation [16] which is calculated by averaging the highest F1-score for the different validations sets. Note that in both probabilistic and data-driven algorithms, the F1-score is optimized as a function of the threshold τ only. In general all the remaining parameters (peak-picking and neural network weights) are tuned beforehand on a separate training set. For the purpose of this paper, we will compare the best-case F1-scores as our focus is the evaluation process and not the algorithms' performance.

In the following, it will be shown that the commonly used evaluation window of 50 ms [1] is not enough for dataset mismatch compensation, when the F1-score is optimized considering only the detection threshold τ. Increasing the evaluation window would lead to a worse resolution. That's why we introduce here the overall shift in annotations δ. While searching for the best threshold, this parameter can be used to find the best match between annotations and detections from the timing point of view. This is done by shifting the annotations for the whole dataset to the right (or left) by adding a small offset δ and searching for the best overlap corresponding to the highest F1-score. In other words, F1-score denoted by F1(τ,δ) is optimized over two parameters: detection threshold τ and overall shift in annotations δ. In our opinion, this doesn't weaken the performance measure. On the contrary, it makes the performance assessment less dependent on the labeling definition or the used method or persons asked to annotate the dataset.

3. EXPERIMENTAL EVALUATION

3.1. Datasets

In order to assess the effect of the overall shift in annotations δ on the F1-score, we used four datasets to which we give the names: MDS, SDS, MAPS_CL, MAPS_AM. The MDS dataset is the one used to evaluate the CNN for onset detection [18]. It is a manually annotated dataset containing audio excerpts from various sources. A 20% portion of this dataset is used for training the network and tuning the peak-picking parameters - except τ - while the remaining examples added to the remaining datasets are used to form four different testing use-cases. The SDS is an automatically annotated dataset [2], containing 138 audio examples. Each example is a mix of 70 notes annotated beforehand - when the notes were isolated - depending on an energy measure. Finally, the MAPS_CL and MAPS_AM are part of the MAPS[1] dataset, specifically the music pieces portion of ENSTDkCl and ENSTDkAm. 'Cl' and 'Am' acronyms refer to *close* and *ambient* way of recordings specifying the distance to the microphone. Those two datasets were recorded using a Yamaha Disklavier and annotations are generated automatically. Table1 gives an overview of the used training and test sets in terms of number of music pieces, onsets and the corresponding number of input points fed to the CNN.

Table 1: Datasets summary

Name	Files #	Onsets #	Points #
MDS_train	77	5979	150K
MDS_test	204	18540	400K
SDS	138	9660	650K
MAPS_CL	30	76364	800K
MAPS_AM	30	77988	800K

3.2. Experiment setup and parameters

For the CNN we used the same characterization as in [16] with minor differences pointed out here. First, an input point is a 3D-tensor (3x15x80) holding 3 magnitude spectrograms with different processing window sizes (23 ms, 46 ms and 93 ms) but same frame rate of 10 ms. The number of logarithmically scaled MEL bands per frame is 80 while 15 is the number of frames per point. A point is given a label '1' if the middle frame is matching an onset and '0.25' for its neighboring points to handle annotations ambiguity. While 50% dropout in training is maintained, no feature normalization is applied which was not necessary for the experiment. Convolutional layers use the ReLU activation functions pushing the CNN to learn spectral differences like the state-of-the-art non-probabilistic ODF's, and the fully-connected layers use the logistic sigmoid. The training is done on mini-batches of 256 points for 150 epochs using the ADAM optimizer [19], minimizing the cross-entropy error. The order of training examples is shuffled after each epoch. This was found to achieve comparable results to the ones reported in [16] when applied on the same datasets with 8-fold cross validation setup and using the same folds.

For the pre-processing, peak-picking and evaluation parameters we stick to the ones used in [16] and [2]. Frames are smoothed with a 50 ms hamming window. For the peak-picking we set θ, α and β to 30 ms, a to 100 ms and b to 70 ms. The evaluation window is kept 50 ms around the ground-truth. Only the probabilistic ODFs were normalized in the range $[0, 1]$ before applying the peak-picking as the algorithms' online capabilities are not our concern in this study.

[1] *Midi Aligned Piano Sounds* dataset - freely available under Creative Commons license

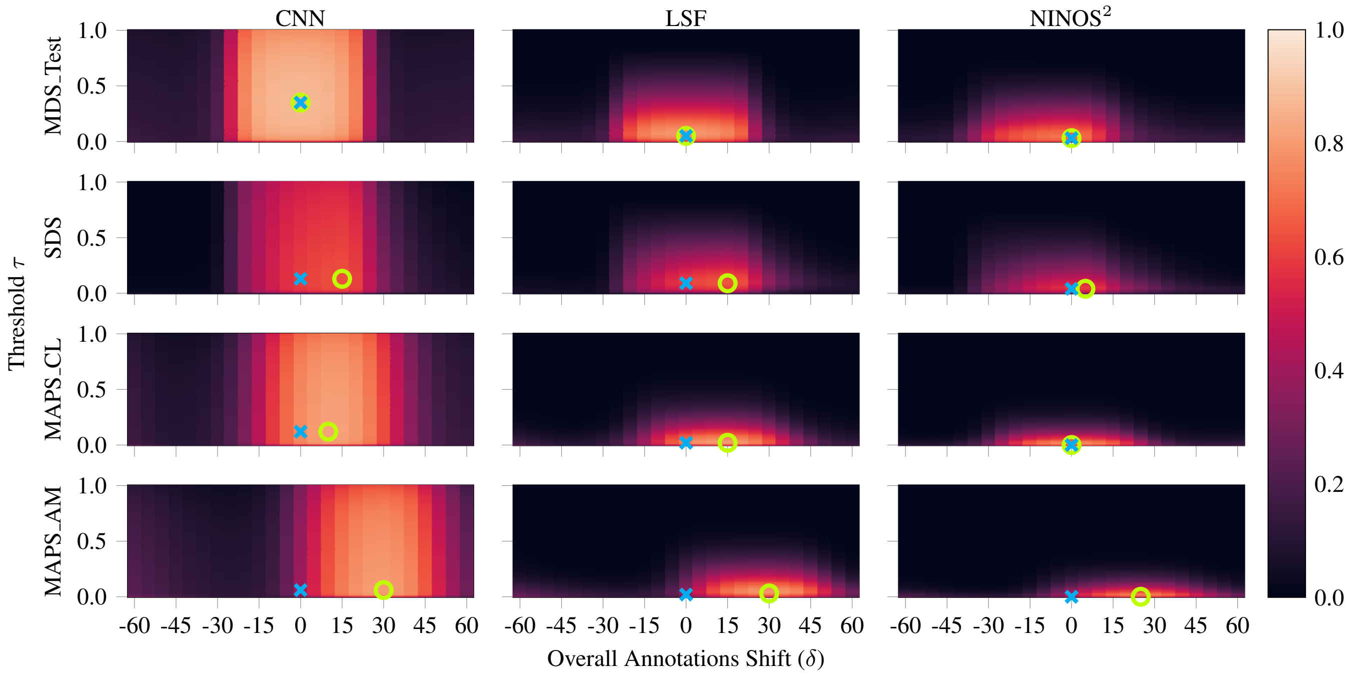

Figure 3: F1-score as a function of threshold τ (vertical axis) and shift δ (horizontal axis), averaged per test set. *Circles* and *crosses* respectively represent the best F1-score average with and without considering the overall shift in annotations δ.

4. RESULTS

In order to study the effect of including the overall shift in annotations δ, we calculated a grid of the average F1-score per dataset as a function of the threshold $\tau \in [0, 1]$ with a step 0.01 vs the shift $\delta \in [-60, 60]$ ms with a step of 5 ms.

Figure 3 shows the threshold-vs-shift F1-score grids for the four test sets using the three ODFs. While the F1-score is less sensitive to threshold variations, it is remarkable how the regions marking the F1-score peaks are not aligned for the different datasets. The F1-score peak values are centered around $\delta = 0$ when the training and testing examples are coming from the same dataset MDS while being always shifted for the other datasets. By comparing the best F1$(\delta, \tau)^*$ to the zero-shift F1$(\delta = 0, \tau^*)$ in Table2, we can see that it is consistently registering a higher F1-score upper bound for the synthetic and disklavier datasets. For instance, neglecting the overall annotations shift for the MAPS_AM dataset reports a best-case performance of more than 50% less than the actual one due to labeling mismatch between the datasets. Note that the shifts $\delta \in [5, 30]$ allowing for a better assessment are calculated on top of the evaluation window, which means a labeling mismatch of up to $30 + 50/2 = 55$ ms which would need an evaluation window of 110 ms to handle it using the methodology in [1]. Different δ values were needed to attain the best scores for the different algorithms. This tells how the different ODFs reply differently to the *seeking* question which confirms the need for a better evaluation independent from the shift.

5. CONCLUSION & FUTURE WORK

This paper argues that the use of an overall shift in annotations parameter δ improves the onset detection evaluation reliability coping with the labeling mismatch between datasets that are annotated differently. δ is crucial for a sound performance assessment making the evaluation process less dependent on the annotation method and thereby more focused on the assessment of the detection algorithm. When reporting the best-case F1-score, we suggest providing the full *F1*(δ, τ) grid. For the on-average F1-score, a possibility is to search for the optimal $(\delta, \tau)^*$ using a validation set taken out from the same dataset as the test set. In the future, we would like to make use of the shift mismatch results to align the examples labeling when mixing together different datasets for training. Moreover, this work can be extended to any MIR time-dependent task.

Table 2: Comparison of best-case F1-score with and without *Overall Shift in Annotations δ Parameter*

		Zero-Shift		Considering the Shift		
	Fold	F1(%)	τ	F1(%)	δ(ms)	τ
CNN	MDS_test	86.7	0.36	86.7	0	0.36
	SDS	59.5	0.14	63.1	15	0.14
	MAPS_CL	75.9	0.13	81.1	10	0.13
	MAPS_AM	39.6	0.05	80.0	30	0.07
LSF	MDS_test	78.4	0.60	78.4	0	0.60
	SDS	59.5	0.10	64.7	15	0.10
	MAPS_CL	72.8	0.03	79.3	15	0.03
	MAPS_AM	36.6	0.03	78.3	30	0.04
NINOS^2	MDS_test	70.2	0.04	70.2	0	0.04
	SDS	57.5	0.05	59.3	5	0.05
	MAPS_CL	76.2	0.01	76.2	0	0.01
	MAPS_AM	45.8	0.01	72.1	25	0.01

6. REFERENCES

[1] A. Lerch and I. Klich, "On the evaluation of automatic onset tracking systems," http://www.musicinformatics.gatech.edu/wp-content_nondefault/uploads/2016/10/Lerch-and-Klich-2005-On-the-Evaluation-of-Automatic-Onset-Tracking-Syst.pdf, zplane.development, Berlin, Tech. Rep., 2005.

[2] M. Mounir, P. Karsmakers, and T. van Waterschoot, "Guitar note onset detection based on a spectral sparsity measure," in *Proc. 24th European Signal Process. Conf. (EUSIPCO 2016)*, 2016, pp. 978–982.

[3] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, "A tutorial on onset detection in music signals," *IEEE Trans. Speech Audio Process.*, vol. 13, pp. 1035–1047, Sept. 2005.

[4] P. Leveau and L. Daudet, "Methodology and tools for the evaluation of automatic onset detection algorithms in music," in *Proc. Int. Symp. Music Information Retrieval (ISMIR)*, 2004, pp. 72–75.

[5] P. Masri and A. Bateman, "Improved modelling of attack transients in music analysis-resynthesis," in *Proc. Int. Comput. Music Conf.*, 1996, pp. 100–103.

[6] J. J. Valero-Mas, E. Benetos, and J. Iñesta, "Assessing the relevance of onset information for note tracking in piano music transcription," in *Proc. 2017 AES Int. Conf. on Semantic Audio*. Audio Engineering Society, 2017.

[7] O. Celma, *Music Recommendation and Discovery*. Springer, 2010.

[8] H. Schreiber and M. Müller, "Accelerating index-based audio identification," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1654–1664, 2014.

[9] A. Wang, "An industrial strength audio search algorithm." in *Proc. Int. Symp. Music Information Retrieval (ISMIR)*, vol. 2003. Washington, DC, 2003, pp. 7–13.

[10] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system with an efficient search strategy," *Journal of New Music Research*, vol. 32, no. 2, pp. 211–221, 2003.

[11] MIREX, "Onset detection results 2018," https://nema.lis.illinois.edu/nema_out/mirex2018/results/aod/summary.html, 2018, accessed 2019-04-11.

[12] S. Böck and G. Widmer, "Local group delay based vibrato and tremolo suppression for onset detection," in *Proc. Int. Symp. Music Information Retrieval (ISMIR)*, 2013, pp. 361–366.

[13] ——, "Maximum filter vibrato suppression for onset detection," in *Proc. 16th Int. Conf. Digital Audio Effects (DAFx-13)*, 2013.

[14] A. Klapuri, "Sound onset detection by applying psychoacoustic knowledge," in *Proc. 1999 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 6. IEEE, 1999, pp. 3089–3092.

[15] P. Masri, "Computer modelling of sound for transformation and synthesis of musical signals," Ph.D. dissertation, University of Bristol, 1996.

[16] J. Schluter and S. Böck, "Improved musical onset detection with convolutional neural networks," in *Proc. 2014 IEEE Int. Conf. Acoustics Speech Signal Process. (ICASSP)*, 2014, pp. 6979–6983.

[17] J. Salamon and J. Urbano, "Current challenges in the evaluation of predominant melody extraction algorithms." in *Proc. Int. Symp. Music Information Retrieval (ISMIR)*, vol. 12, 2012, pp. 289–294.

[18] S. Böck, F. Krebs, and M. Schedl, "Evaluating the online capabilities of onset detection methods," in *Proc. 13th Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2012, pp. 49–54.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.